

IN THE CLAIMS:

- 1 1. (Previously Presented) A method for comparing a first directory comprising unique
2 elements with a second directory comprising unique elements, comprising:
3 (a) for each entry in the first directory, placing a hash value of the entry in a hash
4 table, wherein the first directory is stored on a source storage system;
5 (b) selecting an entry of the second directory, wherein the second directory is lo-
6 cated on a destination storage system;
7 (c) looking up a match between a hash value of the selected entry and the hash
8 value of the entry in the hash table;
9 (d) removing, in response to the match between the hash value of the selected en-
10 try and the hash value of the entry in the hash table, the hash value of the entry from the
11 hash table;
12 (e) determining if an additional second directory entry exists;
13 (f) looping to step (b) in response to identifying the additional second directory
14 entry; and
15 (g) reporting a difference between the first directory and the second directory in
16 response to at least one hash value entry remaining in the hash table.
- 1 2. (Previously Presented) The method of claim 1 further comprising identifying, in re-
2 sponse to not locating the match between the hash value of the selected entry and the
3 hash value of the entry in the hash table, that the hash value of the selected entry is sec-
4 ond directory unique.
- 1 3. (Previously Presented) The method of claim 1 further comprising performing, in re-
2 sponse to not locating the match between the hash value of the selected entry and the
3 hash value of the entry in the hash table, a remedial function.
- 1 4. (Previously Presented) The method of claim 3 wherein the remedial function com-
2 prises deleting the selected entry of the second directory.

- 1 5. (Previously Presented) The method of claim 1 further comprising identifying in re-
2 sponse to no additional entry existing, any remaining hash value entry in the hash table as
3 being first directory unique.
- 1 6. (Previously Presented) The method of claim 1 further comprising performing in re-
2 sponse to no additional entry existing, a remedial function.
- 1 7. (Previously Presented) The method of claim 6 wherein the remedial function com-
2 prises deleting the selected entry of the first directory.
- 1 8. (Previously Presented) The method of claim 6 wherein the remedial function com-
2 prises transferring the selected entry from the first directory to the second directory.
- 1 9. (Previously Presented) The method of claim 1 wherein the data are organized by a
2 RAID system.
- 1 10. (Original) The method of claim 1 wherein the hash table comprises a B-tree.
- 1 11. (Original) The method of claim 1 wherein the hash table comprises a fast lookup data
2 structure.
- 1 12. – 16. (Cancelled)
- 1 17. (Previously Presented) A system for comparing a first data set with a second data set,
2 comprising:
 - 3 (a) means for placing a hash value of each entry of the first data set in a hash ta-
4 ble, wherein the first data set is stored on a source storage system;
 - 5 (b) means for selecting an entry of the second data set, wherein the second data
6 set is located on a destination storage system;

7 (c) means for looking up a match between a hash value of the selected entry and
8 the hash value of the entry in the hash table;

9 (d) means for removing, in response to the match between the hash value of the
10 selected entry and the hash value of the entry in the hash table, the hash value of the entry
11 from the hash table;

12 (e) means for determining if an additional second data set entry exists;

13 (f) means for looping to step (b) in response to identifying the additional second
14 data set entry; and

15 (g) means for reporting a difference between the first data set and the second data
16 set in response to at least one hash value entry remaining in the hash table.

1 18. (Original) The system of claim 17 wherein the hash table comprises a B-tree.

1 19. (Previously Presented) A computer readable medium containing executable program
2 instructions executed by a processor, comprising:

3 (a) for each entry in a first data set, program instructions that place a hash value of
4 the entry in a hash table, wherein the first data set is stored on a source storage system;

5 (b) program instructions that select an entry of the second data set, wherein the
6 second data set is located on a destination storage system;

7 (c) program instructions that look up a match between a hash value of the selected
8 entry and the hash value of the entry in the hash table;

9 (d) program instructions that remove, in response to the match between the hash
10 value of the selected entry and the hash value of the entry in the hash table, the hash
11 value of the entry from the hash table;

12 (e) program instructions that determine if an additional second data set entry ex-
13 ists;

14 (f) program instructions that loop to step (b) in response to identifying the addi-
15 tional second data set entry; and

16 (g) program instructions that report a difference between the first data set and the
17 second data set in response to at least one hash value entry remaining in the hash table.

1 20. (Previously Presented) A method for comparing a first data set with a second data set,
2 comprising:
3 creating a hash table of hashed value entries from the first data set, wherein the
4 first data set is stored on a source storage system;
5 locating a hash value entry from the second data set in the hash table, wherein the
6 second data set is located on a destination storage system;
7 removing, in response to locating the hash value entry from the second data set in
8 the hash table, the hash value entry from the hash table; and
9 recording, in response to at least one hash value entry remaining in the hash table,
10 a difference between the first data set and the second data set.

1 21. (Previously Presented) A method for comparing a first data set with a second data set,
2 comprising:
3 creating a hash table of hash value entries of the first data set, wherein the first
4 data set is stored on a source storage system, and wherein the hash table comprises one or
5 more hashed values of the first data set;
6 determining whether a hashed value of the second data set is identical to a hashed
7 value in the hash table, wherein the second data set is located on a destination storage
8 system;
9 removing, in response to determining that the hashed value of the second data set
10 is identical to the hashed value in the hash table, the identical hashed value from the hash
11 table;
12 recording, in response to determining that the hashed value of the second data set
13 is not identical to the hashed value in the hash table, the hashed value of the second data
14 set as unique; and
15 reporting a difference between the first data set and the second data set in re-
16 sponse to at least one hashed value entry remaining in the hash table.

1 22. (Previously Presented) A method for comparing a first data set with a second data set,
2 comprising:

3 (a) selecting an entry from the first data set;

4 (b) determining if a hashed value of the selected entry of the first data set is in a
5 hash table, wherein the hash table comprises one or more hashed values of the first data
6 set;

7 (c) adding, in response to determining that the hashed value of the selected entry
8 of first data set is not in the hash table, the hashed value of the selected entry of the first
9 data set to the hash table;

10 (d) removing from the hash table, in response to determining that the hashed value
11 of the selected entry of the first data set is in the hash table, the hashed value of the se-
12 lected entry of the first data set;

13 (e) selecting an entry from the second data set;

14 (f) determining if a hashed value of the selected entry of the second data set is in
15 the hash table, wherein the hash table further comprises one or more hashed entries of the
16 second data set;

17 (g) adding, in response to determining that the hashed value of the selected entry
18 of the second data set is not in the hash table, the hashed value of the selected entry of
19 the second data set to the hash table;

20 (h) removing from the hash table, in response to determining that the hashed value
21 of the selected entry of the second data set is in the hash table, the hashed value of the
22 selected entry of the second data set;

23 (i) continuing (a) through (d) and (e) through (h) respectively for all entries in the
24 first and the second data sets until both the first and the second data sets have been com-
25 pletely processed; and

26 (j) reporting a difference between the first data set and the second data set in re-
27 sponse to at least one hashed value remaining in the hash table.

1 23. (Previously Presented) The method of claim 22 wherein the adding the hashed value
2 of the selected entry of the first data set to the hash table further comprises adding infor-

3 mation with the hashed value of the selected entry of the first data set identifying the
4 hashed value of the selected entry of the first data set as originating from the first data set.

1 24. (Previously Presented) The method of claim 22 wherein adding the hashed value of
2 the selected entry of the second data set to the hash table further comprises adding infor-
3 mation with the hashed value of the selected entry of the second data set identifying the
4 hashed value of the selected entry of the second data set as originating from the second
5 data set.

1 25. (Previously Presented) The method of claim 22 wherein the data are organized by a
2 RAID system.

1 26. (Previously Presented) The method of claim 22 further comprising:
2 (k) recording all hashed value entries remaining in the hash table as being unique
3 to either the first data set or the second data set.

1 27. (Original) The method of claim 22 wherein the hash table comprises a B-tree.

1 28. (Original) The method of claim 22 wherein the hash table comprises a fast lookup
2 data structure.

1 29. (Original) The method of claim 22 wherein the first data set comprises a set of direc-
2 tory entries on a source system.

1 30. (Original) The method of claim 22 wherein the second data set comprises a set of di-
2 rectory entries on a destination system.

1 31. (Original) The method of claim 22 wherein the first data set and second data set are
2 on different storage devices.

1 32. (Previously Presented) A system for performing a consistency check of a source di-
2 rectory replicated to a destination directory by comparing entries in the source and desti-
3 nation directories, comprising:

4 one or more storage devices configured to store one or more entries of a group
5 consisting of the source directory and the destination directory; and

6 a process configured to compare entries in the source directory with entries in the
7 destination directory by storing a hash value of each entry of the source directory and the
8 destination directory in a hash table, the process further configured to remove from the
9 hash table any hash value which matches any hash value of the source directory and the
10 destination directory.

1 33. (Original) The system of claim 32 wherein the process executes on a computer asso-
2 ciated with the source directory.

1 34. (Original) The system of claim 32 wherein the process executes on a computer asso-
2 ciated with the destination directory.

1 35. (Cancelled)

1 36. (Previously Presented) A system for performing a consistency check of a source di-
2 rectory and a destination directory, comprising:

3 one or more storage devices configured to store one or more entries of a group
4 comprising of the source directory and the destination directory; and

5 a process configured to compare entries in the source directory with entries in the
6 destination directory by storing a hash value of each entry of the source directory in a
7 hash table, the process further configured to remove any entry from the hash table which
8 matches any hash value of the destination directory.

1 37. (Previously Presented) A computer readable medium containing executable program
2 instructions executed by a processor, comprising:

3 (a) program instructions that select an entry from a first data set, wherein the first
4 data set is stored on a source storage system;

5 (b) program instructions that determine if a hashed value of the selected entry of
6 the first data set is in a hash table, wherein the hash table comprises one or more hashed
7 values of the first data set;

8 (c) program instructions that add, in response to determining that the hashed value
9 of the selected entry of first data set is not in the hash table, the hashed value of the se-
10 lected entry of the first data set to the hash table;

11 (d) program instructions that remove from the hash table, in response to determin-
12 ing that the hashed value of the selected entry of the first data set is in the hash table, the
13 hashed value of the selected entry of the first data set;

14 (e) program instructions that select an entry from a second data set, wherein the
15 second data set is stored on a destination storage system;

16 (f) program instructions that determine if a hashed value of the selected entry of
17 the second data set is in the hash table, wherein the hash table further comprises one or
18 more hashed entries of the second data set;

19 (g) program instructions that add, in response to determining that the hashed value
20 of the selected entry of the second data set is not in the hash table, the hashed value of the
21 selected entry of the second data set to the hash table;

22 (h) program instructions that remove from the hash table, in response to determin-
23 ing that the hashed value of the selected entry of the second data set is in the hash table,
24 the hashed value of the selected entry of the second data set;

25 (i) program instructions that continue (a) through (d) and (e) through (h) respec-
26 tively for all entries in the first and the second data sets until both the first and the second
27 data sets have been completely processed; and

28 (j) program instructions that report a difference between the first data set and the
29 second data set in response to at least one hashed value remaining in the hash table.

- 1 38. (Previously Presented) The computer readable medium of claim 37 further compris-
2 ing program instructions that alternate in selecting entries from the source and destination
3 directories.
- 1 39. (Previously Presented) The method of claim 1 wherein reporting comprises recording
2 the difference on a storage device.
- 1 40. (Previously Presented) The method of claim 22 wherein reporting comprises re-
2 cording the difference on a storage device.
- 1 41. (Previously Presented) The system of claim 32 wherein the process is further config-
2 ured to add to the hash table any hash value which does not match any hash value of the
3 source directory and the destination directory.